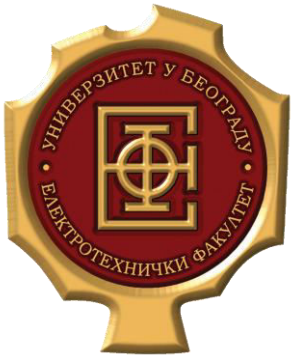
UNIVERZITET U BEOGRADU
ELEKTROTEHNIČKI FAKULTET

Katedra za elektroniku

Predmet: Računarska elektronika



Projekat br. 25: Color picker

Izveštaj

Projekat radili:
Jana Pejanović, 0382/2013
Lazar Dobrijević, 0267/2011

Jun 2018.

Projektni zadatak

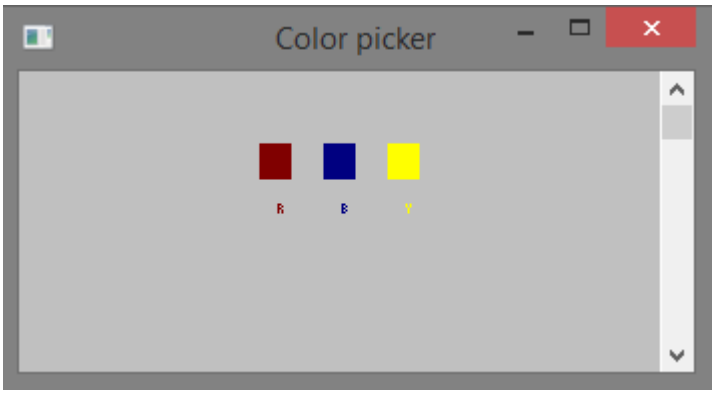
Prvo se na početnom ekranu prikazuju 3 kvadrata u osnovnim bojama, proizvoljne veličine. Ispod kvadrata se nalaze respektivno slova R, B i Y. Od korisnika se tada očekuje unos maksimalno dva slova i tada se prelazi na biranje nijanse. Ukoliko se unese samo jedno slovo, prelazi se na kvadrate koji su u nijansi izabrane boje, s tim da se počinje od bele i završava sa crnom. Ukoliko se unesi 2 slova, ulazi se u novu boju (narandžasta, ljubičasta i zelena) a zatim se ponovo biraju njihove nijanse kako je navedeno.

Realizacija projekta

Izvršavanje programa je realizovano tako da se pri pokretanju programa prikazuje poruka koja korisniku govori o tome kako program funkcioniše i čemu služi, a koja glasi: „Dobrodosli u Color picker. Biranje osnovnih boja se vrsi pritiskanjem tastera R (crvena), B (plava) ili Y (zuta). Biranje sekundarnih boja vrši se pritiskanjem tastera RB (ljubicasta), BY (zelena) ili YR (narandzasta). Biranje nijansi vrsi se pritiskanjem tastera odgovarajuceg broja.“ Ispis poruke na početku programa se vrši pomoću procedure *WriteString*. Da bi se nastavilo izvršavanje programa korisnik treba da pritisne bilo koji taster, što je omogućeno korišćenjem procedure *WaitMsg*.

Nakon što korisnik pritisne bilo koji taster na ekranu se iscrtavaju tri kvadrata, u crvenoj, plavoj i žutoj boji, respektivno. Da bi to bilo moguće potrebno je u programu postaviti handle za ispis podataka pomoću procedure *GetStdHandle*. Nakon toga se čita trenutno stanje kursora pomoću *GetConsoleCursorInfo*, i postavlja novo stanje kursora pomoću *SetConsoleCursorInfo*. Pomoću procedure *SetConsoleTitle* postavlja se ime prozora.

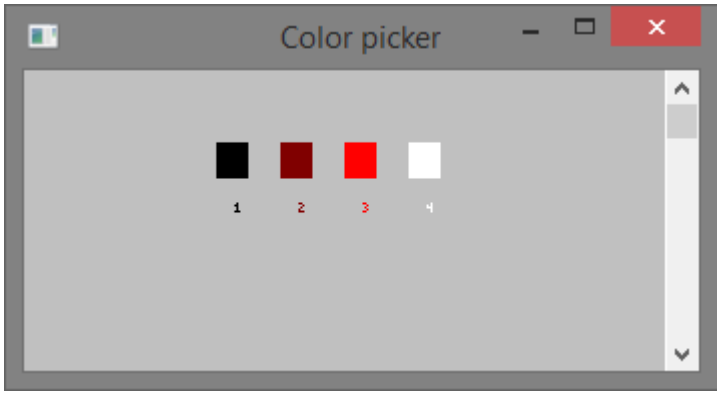
Nakon toga sledi crtanje kvadrata. Registar AX i procedura *SetTextColor* korišćeni su za podešavanje boje teksta, tj. kvadrata. Registar DX korišćen je za čuvanje pozicije koordinata kursora (DL je korišćen za X poziciju, a DH za Y poziciju. Donji deo registra AX, AL, korišćen je za čuvanje ASCII koda karaktera koji se ispisuje. Procedura *Gotoxy* korišćena je za postavljanje kursora na željeno mesto, a procedura *WriteChar* korišćena je za ispisivanje željenog karaktera. Program prvo ispiše slovo ispod kvadrata, a nakon toga crta kvadrat pomoću block karaktera iz ASCII koda. Labele **crveno**, **plavo** i **zuto** su korišćene za podešavanje boje i mesta gde se kvadrat crta, a labele **xinc**, **xdec** i **yinc** su korišćene za crtanje kvadrata, tako što se prvo iscrtaju block karakteri u jednom redu, pa se pređe na sledeći red, i tako dok se ne dođe do poslednjeg reda koji je predviđen za crtanje, za tu proveru korišćen je donji deo registra BX, BL. Na slici 1 prikazan je izgled prozora nakon crtanja RBY kvadrata.



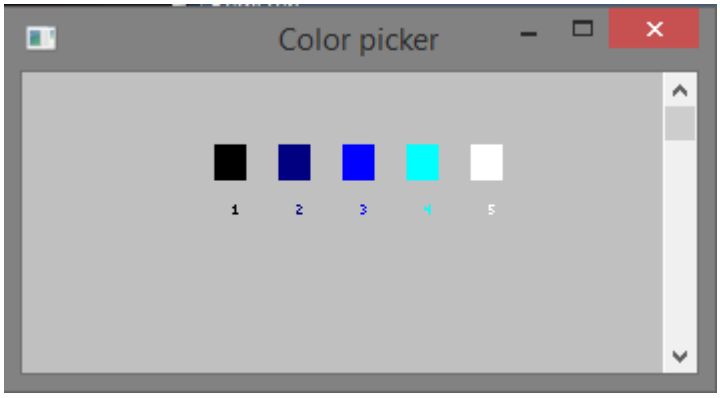
Slika 1. Izgled prozora nakon crtanja RBY kvadrata

Nakon crtanja RBY kvadrata sledi unos slova. Čitanje ulaznog stringa vrši se pomoću procedure *ReadString*, i ako se unese karakter r, proverava se da li je u pitanju kraj stringa, i ako jeste skače se na labelu **crvene_nijanse**, a ako nije proverava se da li je sledeći karakter b, ako jeste skače se na labelu **ljubicaste_nijanse**. Ako se unese karakter b, proverava se da li je u pitanju kraj stringa, i ako jeste skače se na labelu **plave_nijanse** , a ako nije proverava se da li je sledeći karakter y, ako jeste skače se na labelu **zelene_nijanse**. Ako se unese karakter y, proverava se da li je u pitanju kraj stringa, i ako jeste skače se na labelu **zute_nijanse**, a ako nije, proverava se da li je sledeći karakter r, ako jeste skače se na labelu **narandzaste_nijanse**. U slučaju da je uneto više od dva karaktera, ili su uneti pogrešni karakteri potrebno ih je uneti ponovo.

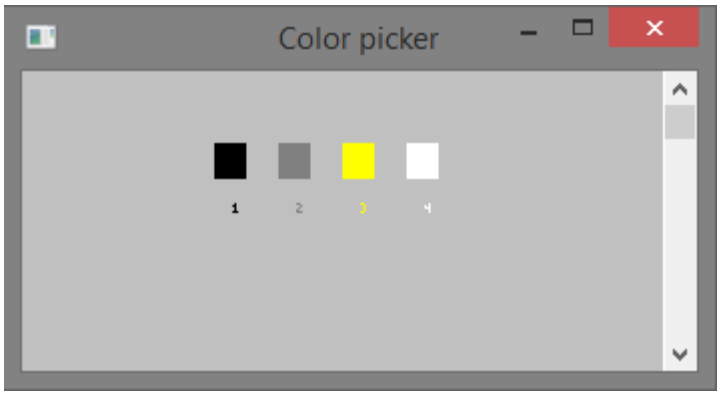
Nakon unosa odgovarajućeg karaktera, na prozoru se iscrtavaju kvadrati u nijansama željene boje, što je urađeno na sličan način kao i crtanje kvadrata u osnovnim bojama. Broj kvadrata zavisi od broja nijansi koje je moguće predstaviti, i prvi kvadrat je uvek crne, a poslednji uvek bele boje. Na sledećim slikama prikazani su izgledi prozora nakon crtanja kvadrata različitih nijansi.



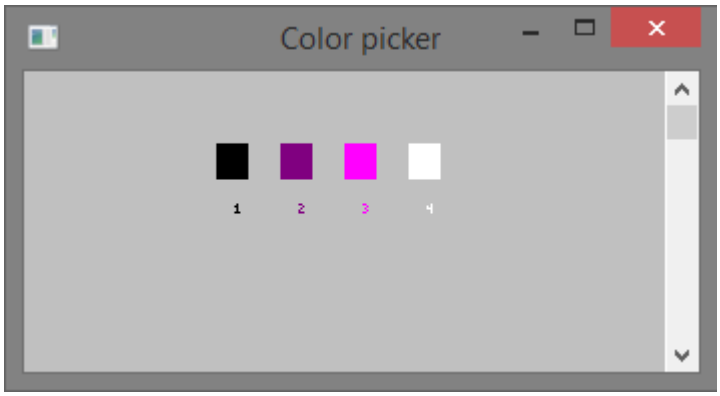
Slika 2. Izgled prozora nakon crtanja kvadrata u crvenoj nijansi



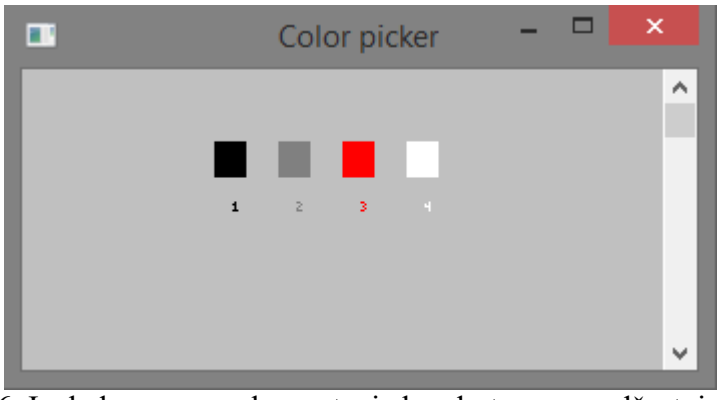
Slika 3. Izgled prozora nakon crtanja kvadrata u plavoj nijansi



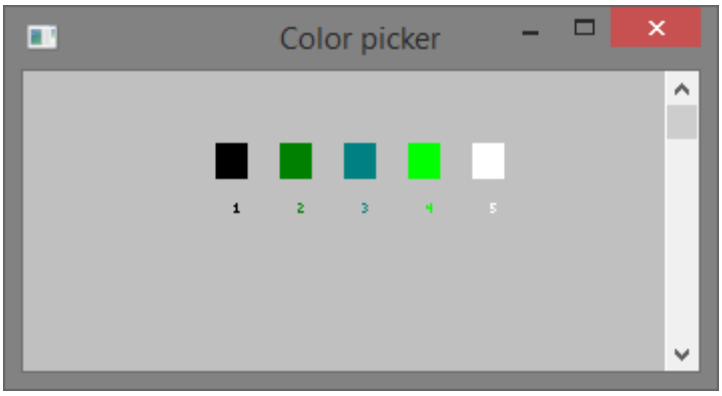
Slika 4. Izgled prozora nakon crtanja kvadrata u žutoj nijansi



Slika 5. Izgled prozora nakon crtanja kvadrata u ljubičastoj nijansi



Slika 6. Izgled prozora nakon crtanja kvadrata u narandžastoj nijansi



Slika 7. Izgled prozora nakon crtanja kvadrata u zelenoj nijansi

Nakon iscrtavanja kvadrata u različitim nijansama od korisnika se očekuje unos broja koji se nalazi ispod kvadrata u čijoj nijansi će biti ispisan tekst skakanjem na labele **unosbrojaN**, gde je N = 1..6. Brojevi se unose pomoću procedure *ReadString* , a zatim se u zavisnosti od pritisnutog broja u registru AX čuva boja u kojoj će tekst biti ispisan.

Ako je unesen ispravan karakter skače se na labelu **ispis**, u kojoj se postavlja odgovarajuća boja teksta pomoću procedure *SetTextColor*, i nakon toga se pomoću procedure *WriteString* ispisuje izlazna poruka: Color Picker. Korisnik izlazi iz programa pritiskom bilo kog tastera.

Izgled koda

```
INCLUDE Irvine32.inc

;-------------------------Promenljive koje se koriste u programu----------------------
.data
por_ulaz byte "Dobrodosli u Color picker.", 0dh, 0ah,
"Biranje osnovnih boja se vrsi pritiskanjem tastera R (crvena), B (plava) ili Y (zuta)",
0dh, 0ah,
"Biranje sekundarnih boja vrši se pritiskanjem tastera RB (ljubicasta), BY (zelena) ili
YR (narandzasta).", 0dh, 0ah,
"Biranje nijansi vrsi se pritiskanjem tastera odgovarajuceg broja.", 0dh, 0ah, 0   ;
String koji opisuje funkciju programa

por_izlaz byte "Color Picker", 0dh, 0ah, 0

wintitle byte "Color picker", 0    ; Naziv programa
cursorInfo CONSOLE_CURSOR_INFO <>  ; Informacije o kursoru

str_size = 4  ; Velicina ulaznog stringa
buffer_size = 3      ; Velicina bafera

.data?
stringIn byte str_size dup(?)       ; Ulazni string u koji se smestaju slova R, B, Y ili
RB, BY, YR
buffer byte buffer_size dup(?)     ; Bafer u koji se smestaju brojevi
stdOutHandle handle ?       ; Handle za ispis podataka
stdInHandle handle ? ; Handle za upis podataka
bytesRead dword ?    ; Broj procitanih bajtova

;-----------------------------------Pocetak programa--------------------------------
.code
```

```
main PROC
; Poruka na pocetku programa
	call Clrscr
	mov eax, black + (lightGray * 16)
	call SetTextColor
	mov  edx,offset por_ulaz
	call WriteString
	call WaitMsg
	call Clrscr

	INVOKE GetStdHandle, STD_OUTPUT_HANDLE    ; Postavlja handle za ispis podataka
	mov  stdOutHandle, eax

	INVOKE GetConsoleCursorInfo, stdOutHandle, ADDR cursorInfo    ; Cita trenutno
stanje kursora
	mov  cursorInfo.bVisible, 0 ; Postavlja vidljivost kursora na nevidljiv
	INVOKE SetConsoleCursorInfo, stdOutHandle, ADDR cursorInfo    ; Postavlja novo
stanje kursora

	INVOKE SetConsoleTitle, ADDR wintitle    ;Postavlja ime prozora

;---------------------------------Crtanje RBY kvadrata-----------------------------

crveno:
	mov ax, red + (lightGray * 16)
	call SetTextColor    ; Podesavanje boje teksta
	mov  dl, 32   ; x pozicija kursora
	mov  dh, 11   ; y pozicija kursora
	mov al, 052h
	call gotoxy
	call writechar       ; ispis slova R
	mov  dl, 30   ; x pozicija kursora
	mov  dh, 6    ; y pozicija kursora
	mov    bl, 9  ; y pozicija kvadrata koja govori o tome da je gotovo crtanje
	mov bh, 061h
	mov ecx, 4    ; broj prolazaka kroz petlju
	mov al, 0DBh
	jmp xinc      ; skok na labelu za crtanje kvadrata

plavo:
	mov ax, blue + (lightGray * 16)
	call SetTextColor
	mov  dl, 40
	mov  dh, 11
	mov al, 042h
	call gotoxy
	call writechar
	mov  dl, 38
	mov  dh, 6
	mov    bl, 9
	mov bh, 062h
	mov ecx, 4
	mov al, 0DBh
	jmp xinc

zuto:
	mov ax, yellow + (lightGray * 16)
	call SetTextColor
```

```
        mov  dl, 48
        mov  dh, 11
        mov al, 059h
        call gotoxy
        call writechar
        mov  dl, 46
        mov  dh, 6
        mov    bl, 9
        mov bh, 063h
        mov ecx, 4
        mov al, 0DBh
        jmp xinc

xinc:
        call gotoxy
        call writechar
        inc dl
        loop xinc

        mov ecx, 4

xdec:
        dec dl
        loop xdec

yinc:
        mov ecx, 4
        inc dh
        cmp bl, dh
        jne xinc

        cmp bh, 061h
        je plavo

        cmp bh,062h
        je zuto

        mov ax, lightGray + (lightGray * 16)
        call SetTextColor

;--------------------------------------Unos slova-------------------------------------

unos_slova:

        mov  edx, OFFSET stringIn
    mov  ecx, str_size
    call ReadString


        .if stringIn == 'r'
               inc edx
               mov al,[edx]
               cmp al, 0
               je crvene_nijanse
               inc edx
               mov ah, [edx]
               cmp ah, 0
               jne unos_slova
```

```
                cmp al,'b'
                je ljubicaste_nijanse
        .elseif stringIn == 'b'
                inc edx
                mov al,[edx]
                cmp al, 0
                je plave_nijanse
                inc edx
                mov ah, [edx]
                cmp ah, 0
                jne unos_slova
                cmp al,'y'
                je zelene_nijanse
        .elseif stringIn == 'y'
                inc edx
                mov al,[edx]
                cmp al, 0
                je zute_nijanse
                inc edx
                mov ah, [edx]
                cmp ah, 0
                jne unos_slova
                cmp al,'r'
                je narandzaste_nijanse
        .endif

        jnz unos_slova

;-------------------------------------Crtanje nijansi---------------------------------

crvene_nijanse:
        call Clrscr
crveno1:
        mov bh, 001h
        mov ax, black + (lightGray * 16)
        call SetTextColor
        jmp kv1

crveno2:
        mov bh, 002h
        mov ax, red + (lightGray * 16)
        call SetTextColor
        jmp kv2

crveno3:
        mov bh, 003h
        mov ax, lightRed + (lightGray * 16)
        call SetTextColor
        jmp kv3

crveno4:
        mov bh, 004h
        mov ax, white + (lightGray * 16)
        call SetTextColor
        jmp kv4

plave_nijanse:
        call Clrscr
```

```
plavo1:
	mov bh, 011h
	mov ax, black + (lightGray * 16)
	call SetTextColor
	jmp kv1

plavo2:
	mov bh, 012h
	mov ax, blue + (lightGray * 16)
	call SetTextColor
	jmp kv2

plavo3:
	mov bh, 013h
	mov ax, lightBlue + (lightGray * 16)
	call SetTextColor
	jmp kv3

plavo4:
	mov bh, 014h
	mov ax, lightCyan + (lightGray * 16)
	call SetTextColor
	jmp kv4

plavo5:
	mov bh, 015h
	mov ax, white + (lightGray * 16)
	call SetTextColor
	jmp kv5

zute_nijanse:
	call Clrscr
zuto1:
	mov bh, 021h
	mov ax, black + (lightGray * 16)
	call SetTextColor
	jmp kv1

zuto2:
	mov bh, 022h
	mov ax, gray + (lightGray * 16)
	call SetTextColor
	jmp kv2

zuto3:
	mov bh, 023h
	mov ax, yellow + (lightGray * 16)
	call SetTextColor
	jmp kv3

zuto4:
	mov bh, 024h
	mov ax, white + (lightGray * 16)
	call SetTextColor
	jmp kv4


narandzaste_nijanse:
```

```
        call Clrscr

narandzasto1:
        mov bh, 031h
        mov ax, black + (lightGray * 16)
        call SetTextColor
        jmp kv1

narandzasto2:
        mov bh, 032h
        mov ax, gray + (lightGray * 16)
        call SetTextColor
        jmp kv2

narandzasto3:
        mov bh, 033h
        mov ax, lightRed + (lightGray * 16)
        call SetTextColor
        jmp kv3

narandzasto4:
        mov bh, 034h
        mov ax, white + (lightGray * 16)
        call SetTextColor
        jmp kv4

zelene_nijanse:

        call Clrscr
zeleno1:
        mov bh, 041h
        mov ax, black + (lightGray * 16)
        call SetTextColor
        jmp kv1

zeleno2:
        mov bh, 042h
        mov ax, green + (lightGray * 16)
        call SetTextColor
        jmp kv2

zeleno3:
        mov bh, 043h
        mov ax, cyan + (lightGray * 16)
        call SetTextColor
        jmp kv3

zeleno4:
        mov bh, 044h
        mov ax, lightGreen + (lightGray * 16)
        call SetTextColor
        jmp kv4

zeleno5:
        mov bh, 045h
        mov ax, white + (lightGray * 16)
        call SetTextColor
        jmp kv5
```

```
ljubicaste_nijanse:
	call Clrscr
ljubicasto1:
	mov bh, 051h
	mov ax, black + (lightGray * 16)
	call SetTextColor
	jmp kv1

ljubicasto2:
	mov bh, 052h
	mov ax, magenta + (lightGray * 16)
	call SetTextColor
	jmp kv2

ljubicasto3:
	mov bh, 053h
	mov ax, lightMagenta + (lightGray * 16)
	call SetTextColor
	jmp kv3

ljubicasto4:
	mov bh, 054h
	mov ax, white + (lightGray * 16)
	call SetTextColor
	jmp kv4

kv1:
	mov  dl, 26
	mov  dh, 11
	mov al, 031h
	call gotoxy
	call writechar
	mov  dl, 24
	mov  dh, 6
	mov    bl, 9
	mov ecx, 4
	mov al, 0DBh
	jmp xinc1


kv2:
	mov  dl, 34
	mov  dh, 11
	mov al, 032h
	call gotoxy
	call writechar
	mov  dl, 32
	mov  dh, 6
	mov    bl, 9
	mov ecx, 4
	mov al, 0DBh
	jmp xinc1


kv3:
	mov  dl, 42
```

```
        mov  dh, 11
        mov al, 033h
        call gotoxy
        call writechar
        mov  dl, 40
        mov  dh, 6
        mov    bl, 9
        mov ecx, 4
        mov al, 0DBh
        jmp xinc1


kv4:
        mov  dl, 50
        mov  dh, 11
        mov al, 034h
        call gotoxy
        call writechar
        mov  dl, 48
        mov  dh, 6
        mov    bl, 9
        mov ecx, 4
        mov al, 0DBh
        jmp xinc1


kv5:
        mov  dl, 58
        mov  dh, 11
        mov al, 035h
        call gotoxy
        call writechar
        mov  dl, 56
        mov  dh, 6
        mov    bl, 9
        mov ecx, 4
        mov al, 0DBh
        jmp xinc1

xinc1:
        call gotoxy
        call writechar
        inc dl
        loop xinc1

        mov ecx, 4

xdec1:
        dec dl
        loop xdec1

yinc1:
        mov ecx, 4
        inc dh
        cmp bl, dh
        jne xinc1

; Crtanje crvenih kvadratica
```

```
        cmp bh, 001h
        je crveno2

        cmp bh, 002h
        je crveno3

        cmp bh, 003h
        je crveno4

        cmp bh, 004h
        je unosbroja1

    ; Crtanje plavih kvadratica
        cmp bh, 011h
        je plavo2

        cmp bh, 012h
        je plavo3

        cmp bh, 013h
        je plavo4

        cmp bh, 014h
        je plavo5

        cmp bh, 015h
        je unosbroja2

    ; Crtanje zutih kvadratica
        cmp bh, 021h
        je zuto2

        cmp bh, 022h
        je zuto3

        cmp bh, 023h
        je zuto4

        cmp bh, 024h
        je unosbroja3

    ; Crtanje narandzastih kvadratica
        cmp bh, 031h
        je narandzasto2

        cmp bh, 032h
        je narandzasto3

        cmp bh, 033h
        je narandzasto4

        cmp bh, 034h
        je unosbroja4

    ; Crtanje zelenih kvadratica
        cmp bh, 041h
        je zeleno2
```

```
        cmp bh, 042h
        je zeleno3

        cmp bh, 043h
        je zeleno4

        cmp bh, 044h
        je zeleno5

        cmp bh, 045h
        je unosbroja5

; Crtanje ljubicastih kvadratica
        cmp bh, 051h
        je ljubicasto2

        cmp bh, 052h
        je ljubicasto3

        cmp bh, 053h
        je ljubicasto4

        cmp bh, 054h
        je unosbroja6

;---------------------------------------Unos broja----------------------------------
unosbroja1:
        mov ax, lightGray + (lightGray * 16)
        call SetTextColor

        mov  edx, OFFSET buffer
        mov  ecx, buffer_size
        call ReadString

        .if buffer == '1'
                inc edx
                mov al, [edx]
                cmp al, 0
                jne unosbroja1
                mov ax, black + (lightGray * 16)
        .elseif buffer == '2'
                inc edx
                mov al, [edx]
                cmp al, 0
                jne unosbroja1
                mov ax, red + (lightGray * 16)
        .elseif buffer == '3'
                inc edx
                mov al, [edx]
                cmp al, 0
                jne unosbroja1
                mov ax, lightRed + (lightGray * 16)
        .elseif buffer == '4'
                inc edx
                mov al, [edx]
                cmp al, 0
                jne unosbroja1
                mov ax, white + (lightGray * 16)
```

```
        .endif

        jnz unosbroja1
        jmp ispis

unosbroja2:
        mov ax, lightGray + (lightGray * 16)
        call SetTextColor

        mov  edx, OFFSET buffer
        mov  ecx, buffer_size
        call ReadString

        .if buffer == '1'
                inc edx
                mov al, [edx]
                cmp al, 0
                jne unosbroja2
                mov ax, black + (lightGray * 16)
        .elseif buffer == '2'
                inc edx
                mov al, [edx]
                cmp al, 0
                jne unosbroja2
                mov ax, blue + (lightGray * 16)
        .elseif buffer == '3'
                inc edx
                mov al, [edx]
                cmp al, 0
                jne unosbroja2
                mov ax, lightBlue + (lightGray * 16)
        .elseif buffer == '4'
                inc edx
                mov al, [edx]
                cmp al, 0
                jne unosbroja2
                mov ax, lightCyan + (lightGray * 16)
        .elseif buffer == '5'
                inc edx
                mov al, [edx]
                cmp al, 0
                jne unosbroja2
                mov ax, white + (lightGray * 16)
        .endif

        jnz unosbroja2
        jmp ispis


unosbroja3:
        mov ax, lightGray + (lightGray * 16)
        call SetTextColor

        mov  edx, OFFSET buffer
        mov  ecx, buffer_size
        call ReadString

        .if buffer == '1'
```

```
		inc edx
		mov al, [edx]
		cmp al, 0
		jne unosbroja3
		mov ax, black + (lightGray * 16)
	.elseif buffer == '2'
		inc edx
		mov al, [edx]
		cmp al, 0
		jne unosbroja3
		mov ax, gray + (lightGray * 16)
	.elseif buffer == '3'
		inc edx
		mov al, [edx]
		cmp al, 0
		jne unosbroja3
		mov ax, yellow + (lightGray * 16)
	.elseif buffer == '4'
		inc edx
		mov al, [edx]
		cmp al, 0
		jne unosbroja3
		mov ax, white + (lightGray * 16)
	.endif

	jnz unosbroja3
	jmp ispis

unosbroja4:
	mov ax, lightGray + (lightGray * 16)
	call SetTextColor

	mov  edx, OFFSET buffer
	mov  ecx, buffer_size
	call ReadString

	.if buffer == '1'
		inc edx
		mov al, [edx]
		cmp al, 0
		jne unosbroja4
		mov ax, black + (lightGray * 16)
	.elseif buffer == '2'
		inc edx
		mov al, [edx]
		cmp al, 0
		jne unosbroja4
		mov ax, gray + (lightGray * 16)
	.elseif buffer == '3'
		inc edx
		mov al, [edx]
		cmp al, 0
		jne unosbroja4
		mov ax, lightRed + (lightGray * 16)
	.elseif buffer == '4'
		inc edx
		mov al, [edx]
		cmp al, 0
```

```
                jne unosbroja4
                mov ax, white + (lightGray * 16)
        .endif

        jnz unosbroja4
        jmp ispis

unosbroja5:
        mov ax, lightGray + (lightGray * 16)
        call SetTextColor

        mov  edx, OFFSET buffer
        mov  ecx, buffer_size
        call ReadString

        .if buffer == '1'
                inc edx
                mov al, [edx]
                cmp al, 0
                jne unosbroja5
                mov ax, black + (lightGray * 16)
        .elseif buffer == '2'
                inc edx
                mov al, [edx]
                cmp al, 0
                jne unosbroja5
                mov ax, green + (lightGray * 16)
        .elseif buffer == '3'
                inc edx
                mov al, [edx]
                cmp al, 0
                jne unosbroja5
                mov ax, cyan + (lightGray * 16)
        .elseif buffer == '4'
                inc edx
                mov al, [edx]
                cmp al, 0
                jne unosbroja5
                mov ax, lightGreen + (lightGray * 16)
        .elseif buffer == '5'
                inc edx
                mov al, [edx]
                cmp al, 0
                jne unosbroja5
                mov ax, white + (lightGray * 16)
        .endif

        jnz unosbroja5
        jmp ispis

unosbroja6:
        mov ax, lightGray + (lightGray * 16)
        call SetTextColor

        mov  edx, OFFSET buffer
        mov  ecx, buffer_size
        call ReadString
```

```
	.if buffer == '1'
		inc edx
		mov al, [edx]
		cmp al, 0
		jne unosbroja6
		mov ax, black + (lightGray * 16)
	.elseif buffer == '2'
		inc edx
		mov al, [edx]
		cmp al, 0
		jne unosbroja6
		mov ax, magenta + (lightGray * 16)
	.elseif buffer == '3'
		inc edx
		mov al, [edx]
		cmp al, 0
		jne unosbroja6
		mov ax, lightMagenta + (lightGray * 16)
	.elseif buffer == '4'
		inc edx
		mov al, [edx]
		cmp al, 0
		jne unosbroja6
		mov ax, white + (lightGray * 16)
	.endif

	jnz unosbroja6
	jmp ispis

;-------------------------------------Ispis teksta---------------------------------
ispis:

	call SetTextColor
	call Clrscr
	mov  edx,offset por_izlaz
	call WriteString
	call WaitMsg
	call Clrscr

;-----------------------------------------Izlaz--------------------------------------
INVOKE ExitProcess,0

main ENDP
END main
```